CLAIMS

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A method for virtualizing access to named system objects, the method comprising the steps of:
receiving a request to access a system object from a process executing in the context of a user isolation scope, the request including a virtual name for the system object;
determining a rule associated with the request;
forming a literal name for the system object in response to the determined rule; and
issuing to the operating system a request to access the system object, the request including the literal name for the system object.

2. The method of claim 1 wherein step (a) comprises receiving a request from a process executing in the context of a user isolation scope to access a system object selected from the group consisting of a semaphore, a mutex, a mutant, a timer, an event, a job object, a file-mapping object, a

section, a named pipe, and a mailslot, the request including a virtual name for the system object.

3. The method of claim 1 wherein step (a) comprises intercepting a request to access a system object from a process executing in the context of a user isolation scope, the request including a virtual name for the system object.

4. The method of claim 1 wherein step (a) comprises receiving a request from a process executing in the context of a user isolation scope to open a system object, the request including a virtual name for the system object.

5. The method of claim 1 wherein step (a) comprises receiving a request from a process executing in the context of a user isolation scope to create a system object, the request including a virtual name for the system object.

6. The method of claim 1 wherein step (b) comprises determining that a rule action selected from the group consisting of ignore, redirect and isolate, is associated with the request.

7. The method of claim 1 wherein step (b) comprises accessing a rules engine to determine a rule action associated with the virtual name included in the received request.

8. The method of claim 1 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier.

9. The method of claim 8 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the process making the request is associated.

10. The method of claim 8 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the user isolation scope in which the process making the request executes.

11. The method of claim 1 wherein step (c) further comprises the step of forming a literal name for the system object identifying the system object as having global visibility.

12. The method of claim 1 wherein step (c) further comprises the step of forming a literal name for the system object identifying the system object as having session visibility.

13. The method of claim 1 wherein step (c) comprises forming a literal name for the system object that is substantially identical to the virtual name provided in the request.

14. The method of claim 1 further comprising the step of receiving a handle from the operating system identifying the accessed object.

15. The method of claim 14 further comprising the step of transmitting the handle to the process.

16. The method of claim 1 further comprising the step of receiving a request to access the system object from a second process executing in the context of a second user isolation scope, the request including the virtual name for the object.

17. The method of claim 16 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier.

18. The method of claim 17 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the process making the request is associated.

19. The method of claim 17 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the second user isolation scope in which the process making the request executes.

20. The method of claim 16 wherein step (c) comprises forming a literal name for the system object that is substantially identical to the virtual name provided in the request.

21. The method of claim 1 further comprising the step of receiving a request to access the system object from a second process executing in the context of the user isolation scope, the request including the virtual name for the object.

22. The method of claim 21 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier.

23. The method of claim 22 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the second process making the request is associated.

24. The method of claim 22 wherein step (c) comprises forming a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the user isolation scope in which the second process making the request executes.

25. The method of claim 21 wherein step (c) comprises forming a literal name for the system object that is substantially identical to the virtual name provided in the request.

26. An apparatus for virtualizing access to named system objects comprising:

a hooking mechanism receiving a request to access a system object from a process executing in the context of a user isolation scope, the request including a virtual name for the system object;

a name virtualization engine forming a literal name for the system object; and

an operating system interface requesting access to the system object using the literal name.

27. The apparatus of claim 26 wherein the hooking mechanism intercepts a request to open a system object.

28. The apparatus of claim 26 wherein the hooking mechanism intercepts a request to create a system object

29. The apparatus of claim 26 further comprising a rules engine storing a rule associated with the request.

30. The apparatus of claim 29 wherein the rules engine comprises a database.

31. The apparatus of claim 26 wherein the name virtualization engine forms a literal name for the system object that is substantially identical to the virtual name.

32. The apparatus of claim 26 wherein the name virtualization engine forms a literal name for the system object using the virtual name and a scope-specific identifier.

33. The apparatus of claim 32 wherein the scope-specific identifier is associated with an application isolation scope with which the process making the request is associated.

34. The method of claim 32 wherein the scope-specific identifier is associated with the user isolation scope in which the process making the request executes.